A Methodology for Agile Development of System Security Architectures in Complex Systems

Ronda Henning

rhenning@harris.com

Introduction

A system is an interconnected set of elements that is coherently organized in a way that achieves something.

Donella H. Meadows
Thinking in Systems

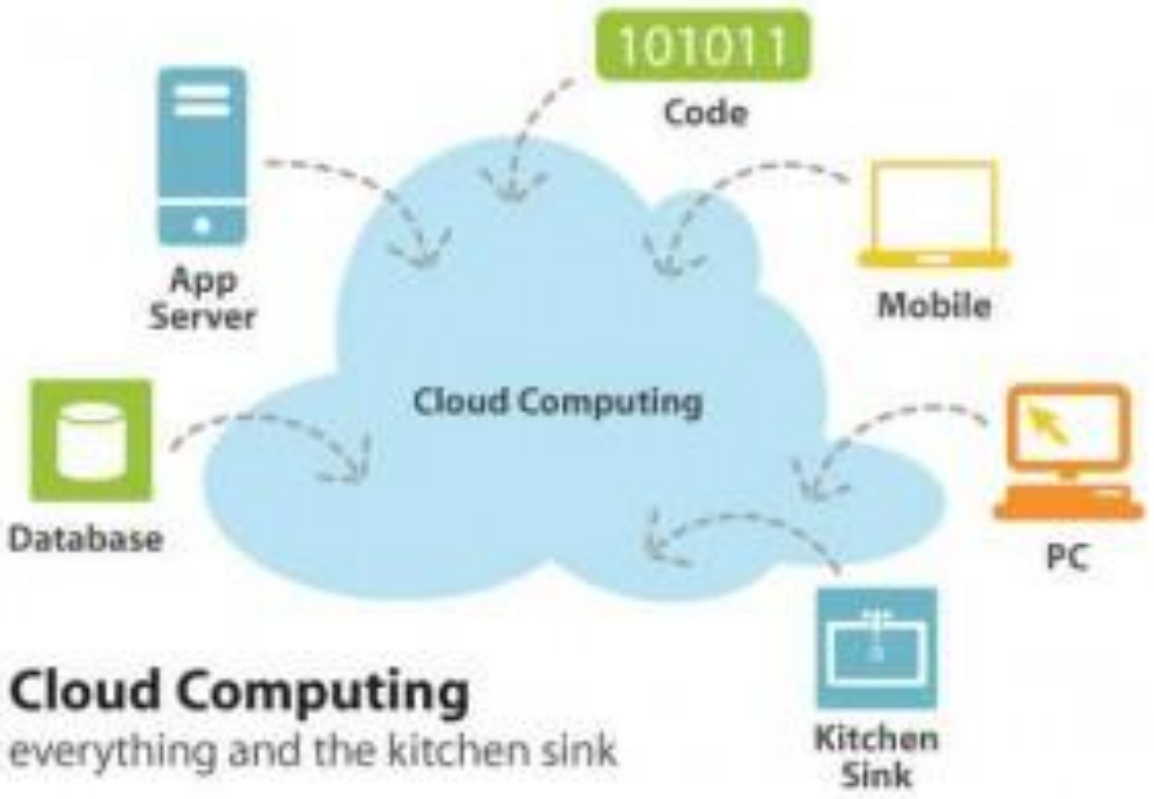
System Principles

- More than the sum of its parts
- Many of the interconnections in systems operate through the flow of information.
- The least obvious part of the system, its function or purpose, is often the most crucial determinant of the system’s behavior.
- System Structure is the source of system behavior.
- System behavior reveals itself as a series of events over time.

The World of Security….

- Normally seen as metadata about data
    - – Who can access data – by group membership, role, explicit ID
        - Under what circumstances – temporal or location
  - – Confidentiality, integrity, availability constraints
  - – Information pedigree
  - – Constrains information flow

"The Cloud" Adds Complexity

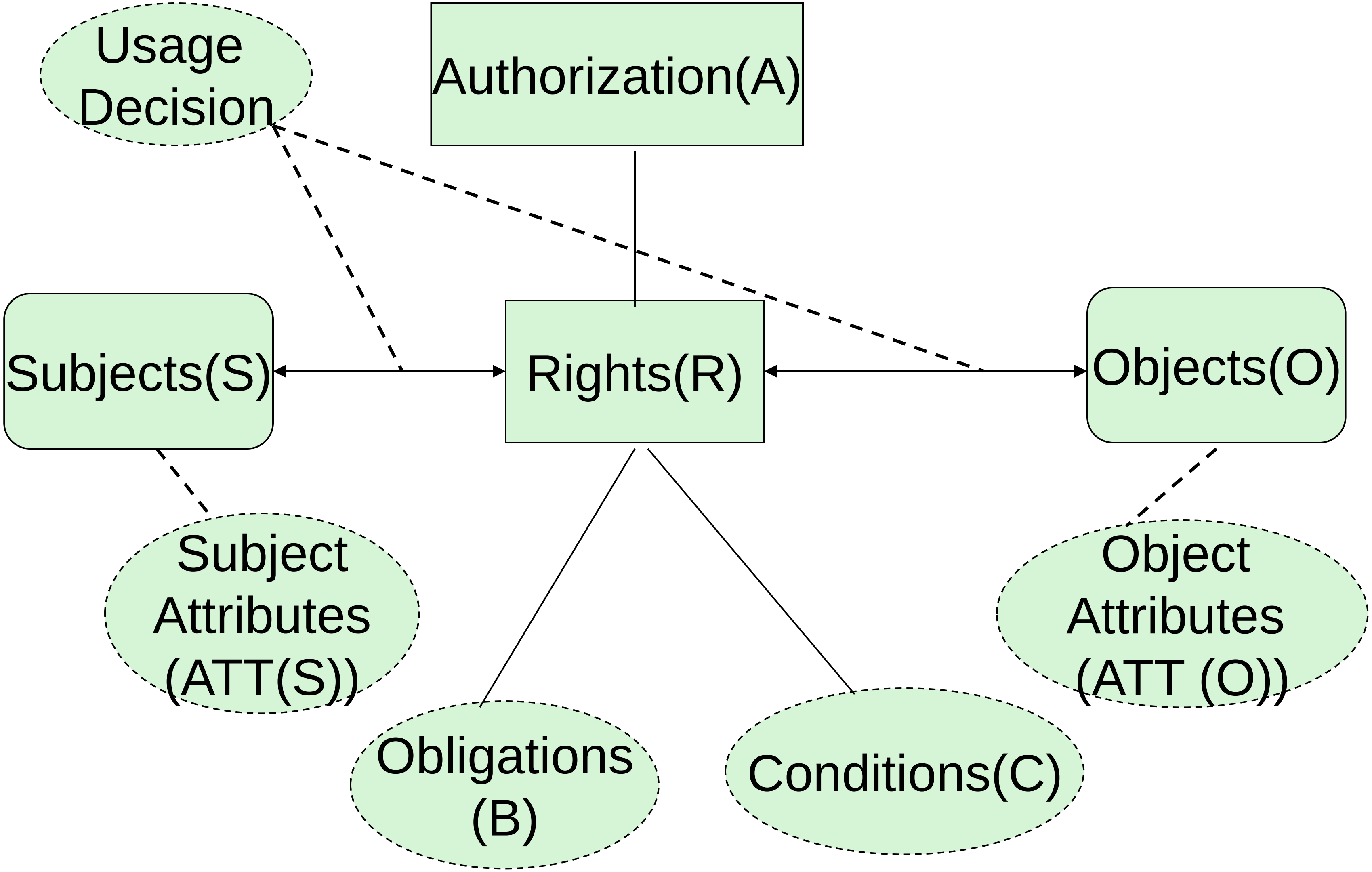


But it also reflects reality

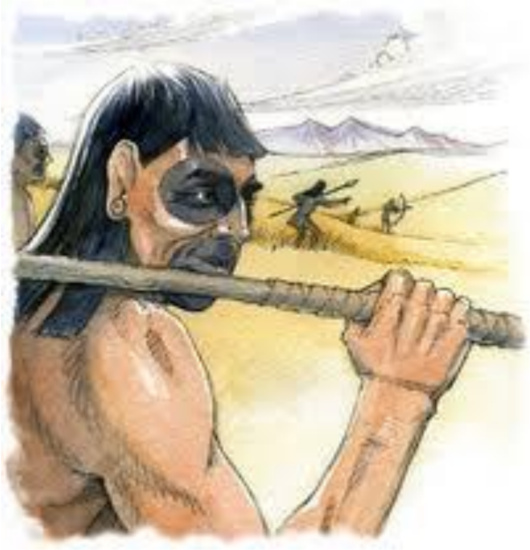
The problem

- Explaining security is painful
  - <pathname>classification >= <User authorization>
  - Modelers want a formal, logically proven policy
  - Formal methods do not accommodate human behavior very well
  - Exceptions are not tolerated
- If you can’t explain it, how do you know you’ve correctly implemented it

For Example

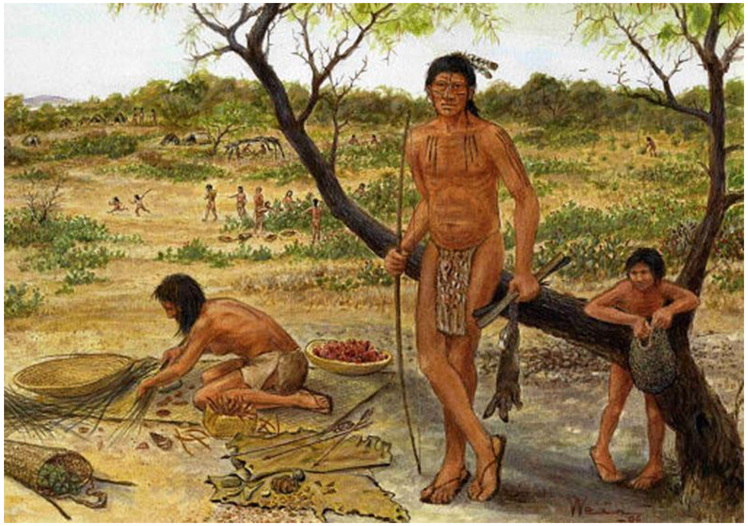

Usage Decision
Authorization(A)
Subjects(S)
Rights(R)
Objects(O)
Subject Attributes (ATT(S))
Object Attributes (ATT (O))
Obligations (B)
Conditions(C)

An observation….



Man as independent creature

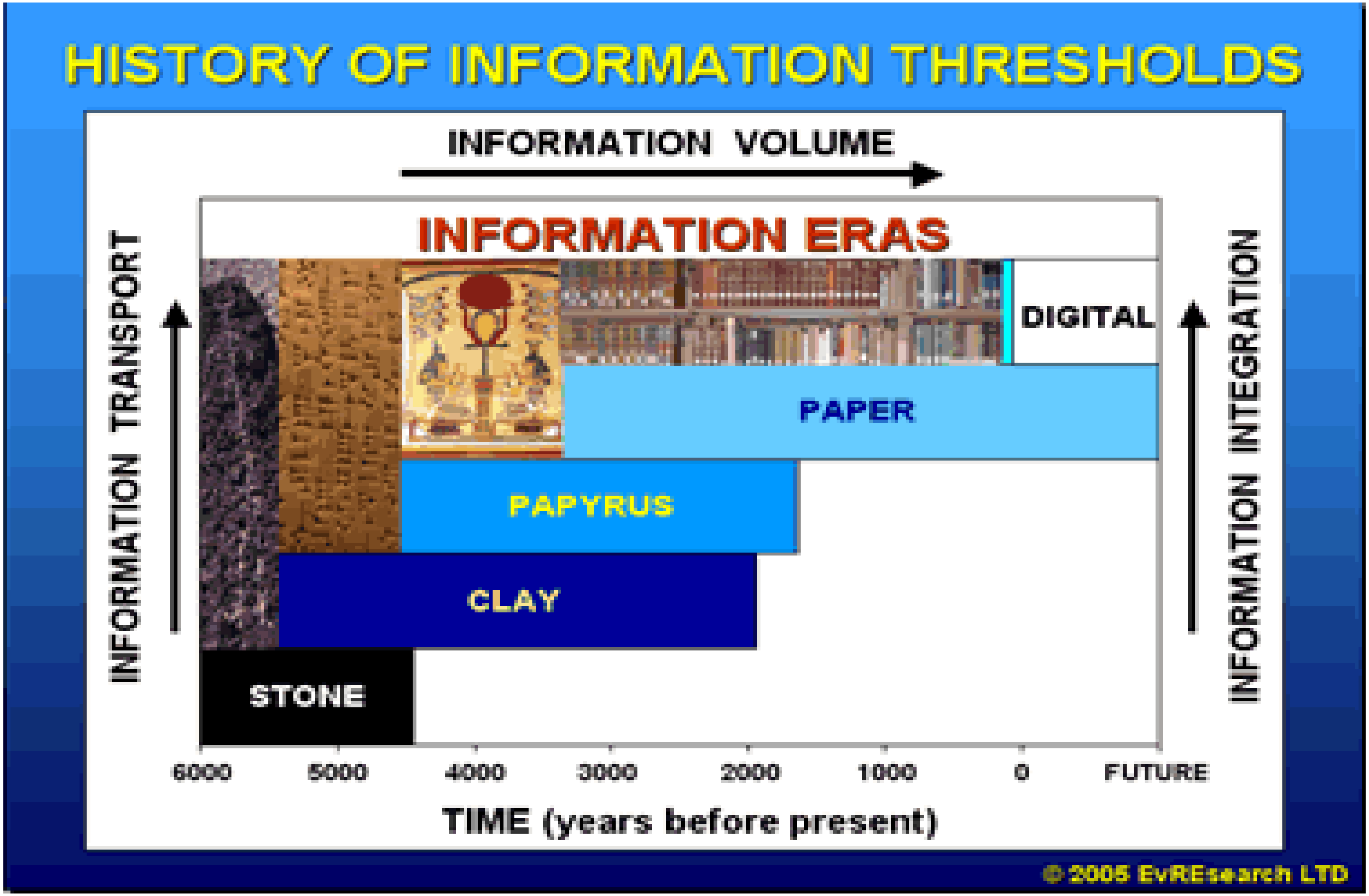
- •Foraged for own food
- •No concept of specialization
- •Little shared knowledge



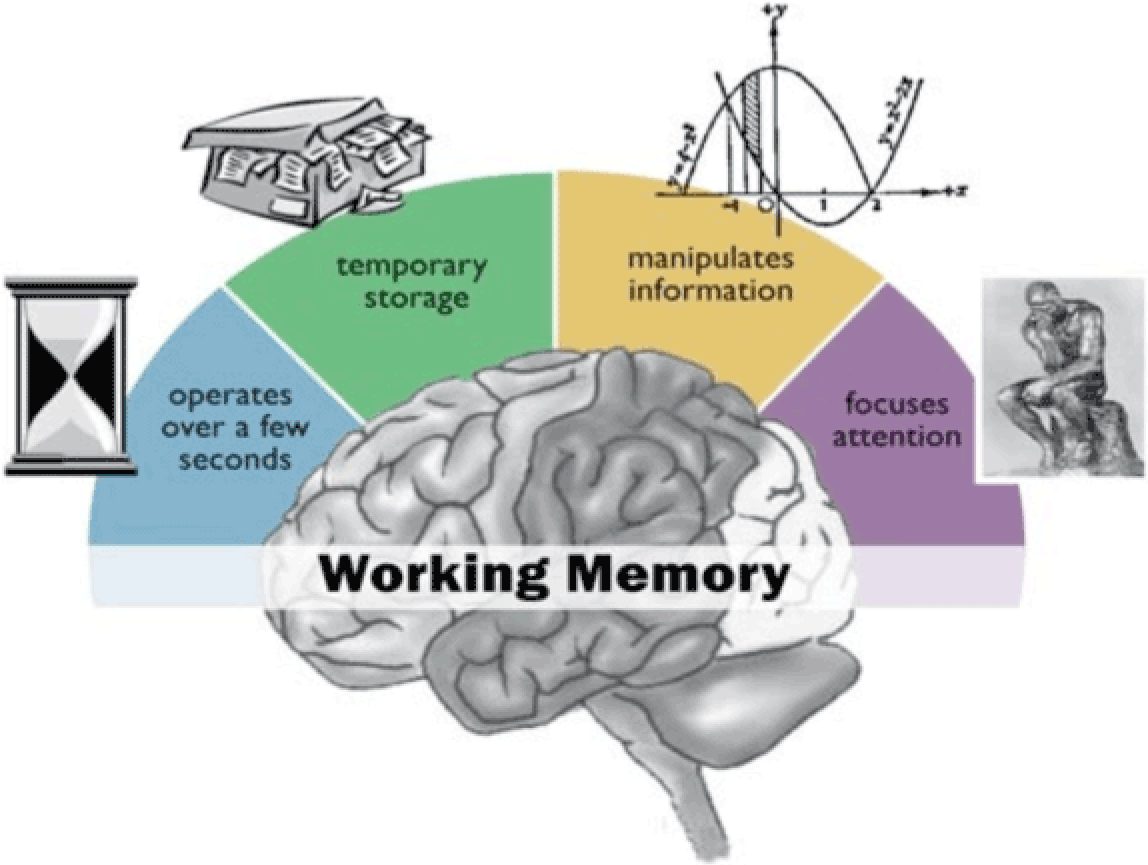
Man as hunter gatherer

- Simple specialization
- Concept of “tribes”
- Shared knowledge
  - Water
  - Plants
  - Territory

Information Portability

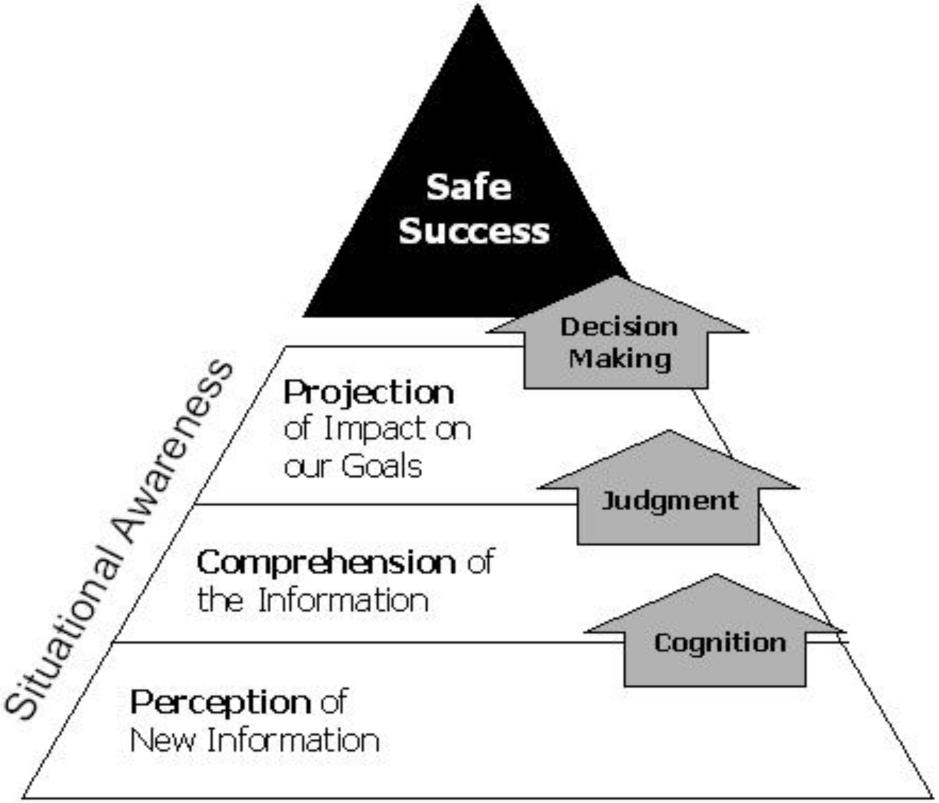


Memory

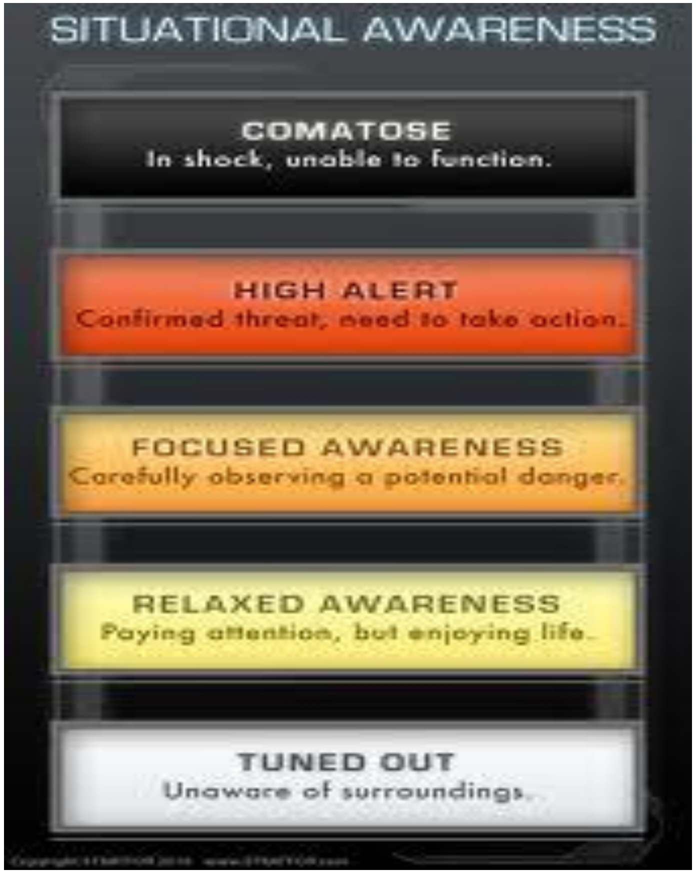


The mind is an efficient guesser,
using context and content to create and evaluate alternatives and select the most probable answer

Models of situational awareness



Safe Success
Decision Making
Projection of Impact on our Goals
Judgment
Comprehension of the Information
Cognition
Perception of New Information
Situational Awareness



SITUATIONAL AWARENESS

COMATOSE
In shock, unable to function.

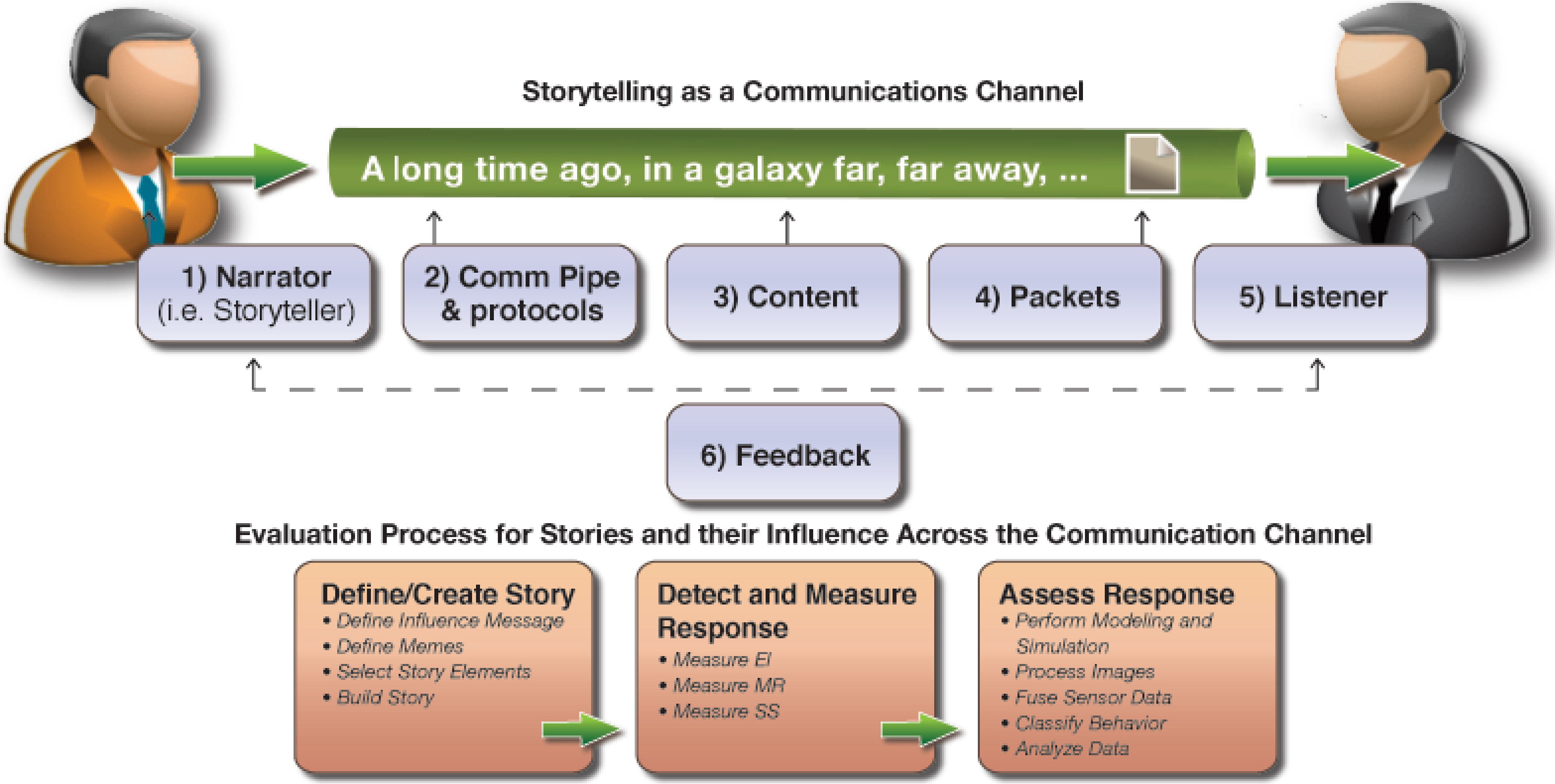
HIGH ALERT
Confirmed threat, need to take action.

FOCUSED AWARENESS
Carefully observing a potential danger.

RELAXED AWARENESS
Paying attention, but enjoying life.

TUNED OUT
Unaware of surroundings.

NeuroCognitive Story Model



Storytelling as a Communications Channel

A long time ago, in a galaxy far, far away, ...

1) Narrator (i.e. Storyteller)

2) Comm Pipe & protocols

3) Content

4) Packets

5) Listener

6) Feedback

Evaluation Process for Stories and their Influence Across the Communication Channel

Define/Create Story
- *Define Influence Message*
- *Define Memes*
- *Select Story Elements*
- *Build Story*

Detect and Measure Response
- *Measure EI*
- *Measure MR*
- *Measure SS*

Assess Response
- *Perform Modeling and Simulation*
- *Process Images*
- *Fuse Sensor Data*
- *Classify Behavior*
- *Analyze Data*

The Methodology

- Use story to explain security policy
- Sentences are composed of subject, verb, object
- The brain is wired to understand story, not proofs
- An alternative approach to capture context

- To access the terrain map, a user must be in theater, with a need to know for the resolution required. Exception: Team mission planning….

Rationale

- Expressing human comprehensible policy…
  - Reduces mis-understanding
  - Disconnects between the user, the developer, and the evaluator/tester
  - Captures “intent” and context for use

What it’s not

- A use case:
  - One use case to capture all security requirements would be very long
  - Defeats the purpose of “building security in” by segregating the function
- A story in the agile sense
  - Reflects the functionality to be implemented in a given scrum
  - This reflects the end state functionality

What it provides

- Flexibility
  - Implementation independent
  - Captures what the mechanism has to do
  - Accommodates architecture substitutions
- Traceability to stakeholder requirements
- A “contract” for system behavior
  - Expressed in language that user can understand
  - Translated via traceability and decomposition into actual mechanisms and architectural allocation

Conclusion

- Narrative information is most readily processed by human cognitive functions
- Taking advantage of structured English facilitates understanding of desired behaviors
- Minimizes requirement disconnects
  - “Right size” desired assurance